Exploiting HPC for HEP workloads

- Rod Walker, LMU Munich
1st Feb 2016

- ATLAS & Munich centric view
- Short-term: this and maybe next generation HPC
- General comments on HPC for HEP

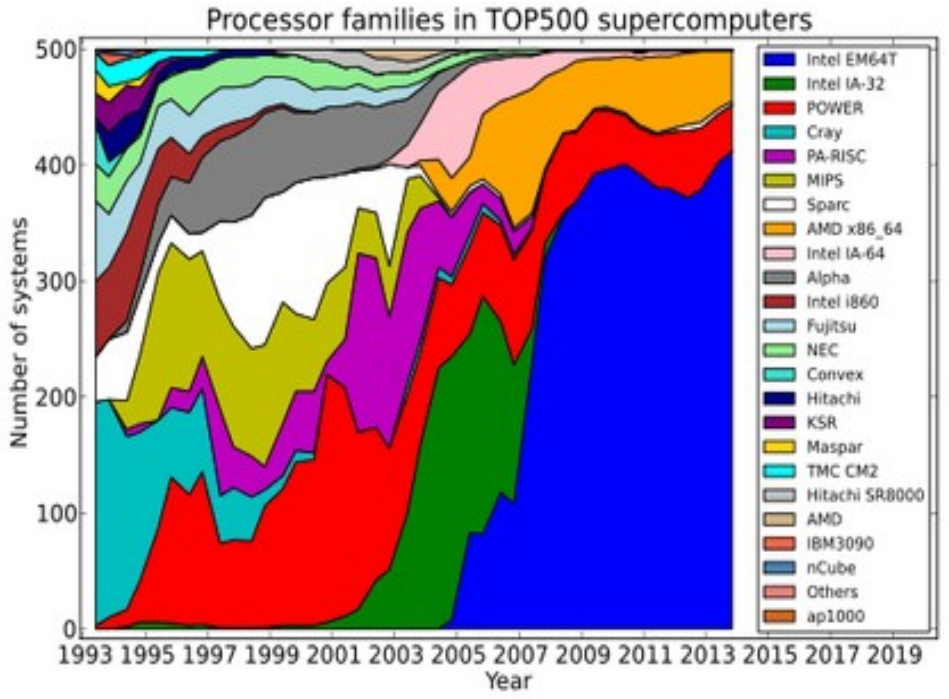
HEP Workloads

- Geant4 simulation
  - deeply bound to experiment frameworks
    - only built for x86 & linux
    - some attempt to make stand-alone to build anywhere
  - Multi-core works. Multi-threaded for MIC maybe.
  - ATLAS Event Service allows short/preemptable jobs
    - bookkeeping exercise to only lose currently processed event when preempted
- Reconstruction
  - built only for SL6. Requires conditions data
  - next generation GAUDI-Hive multi-threaded
- Event generators
  - some are cpu intensive and stand-alone
    - can be built and optimized on other architectures, Alpgen

Types of HPC

- Huge simplification …..
  - x86 linux with or without accelerator(GPU/MIC)
  - PowerPC

Not Flops



Munich HPC

- LRZ SuperMUC
  - Phase 1: 150k cores, Sandybridge
  - Phase 2: 86k cores, Haswell
  - ATLAS has 20Mcore allocation
    - effectively open-ended allocation if preempt-only
- Max Planck Institute computer centre: Hydra
  - 83k Sandybridge

ATLAS ProdSys integration

- Benefit from Nordugrid middleware and experience
- Pilot model no longer flies – no IP
  - submit pre-loaded pilots
- ARC CE designed for non-intrusive integration
  - stage-in/out data on shared FS, BS interface(LoadLeveler)
  - added ability to have remote CE access cluster via ssh
- ATLAS SW available by rsync of cvmfs and relocation, more recently via parrot-cvmfs.
  - no outbound IP → no Frontier → only sim
  - only whole-node scheduled → AthenaMP

ARC CE via ssh

- Not allowed a service on HPC login node
- Key-base ssh is allowed
- Mount shared FS using Fuse(sshfs)
- Interact with BS using ssh to run commands
  - important details solved by Michi(Bern, for CSCS)
- Remarkably stable
- HPC Cluster has gateway outside their control
  - on VM at LMU – data transfer path not optimal, scaling
  - HPC should provide ARC CE

Software: Parrot-CVMFS for HPC

- CVMFS needs no introduction
  - needs a local cache,... and Stratum-0 source
  - needs WN root mount, or at least FUSE
  - needs outbound IP connectivity
- HPC fails on all counts
  - no local disk, no (local)cache
  - no root, no fuse
  - no connectivity

Parrot-cvmfs

- Parrot is part of the cctools suite
  - http://ccl.cse.nd.edu/software/
  - much history and collaboration with cvmfs(Blomer)
- Wrapper around command/script/binary to intercept FS operations and do something
  - inc. HTTP, FTP, GridFTP, iRODS, CVMFS, Chirp
  - access to /cvmfs handled by plugin from Jakob
- Still requires outbound IP and proxy.

Parrot fun

Cvmfs anywhere

```
[aipanda121] cctools $ ls /cvmfs/atlas.cern.ch
ls: cannot access /cvmfs/atlas.cern.ch: No such file or directory
[aipanda121] cctools $ cctools-5.3.4-x86_64-redhat6/bin/parrot_run bash
[aipanda121] cctools $ ls /cvmfs/atlas.cern.ch
repo
[aipanda121] cctools $
```

Test grid job without AFS

```
[aipanda121] cctools $ ls -d /afs/cern.ch
/afs/cern.ch
[aipanda121] cctools $ cctools-5.3.4-x86_64-redhat6/bin/parrot_run --mount=/afs=/dummy bash
bash-4.1$ ls -d /afs/cern.ch
ls: cannot access /afs/cern.ch: No such file or directory
bash-4.1$
```

Parrot alien cache

- Cvmfs cache can be on a shared FS
  - used by all clients, but still needs outbound IP
- Cvmfs cache can be pre-loaded
  - copy of stratum-0, 100% cache hits
  - no outbound IP required → HPC
- Pre-loading can choose directories
  - anything containing .cvmfscatalog file
  - eg. base releases, DBReleases
  - faster than rsync
- Parrot ptrace style intercepts not without difficulty
  - several problems found and quickly fixed by cctools dev
    - argument ignored, seg fault, tar for log fails (on SLES)

```
> export PARROT_CVMFS_ALIEN_CACHE=/gpfs/work/pr58be/ri32buz2/cvmfs_preload
```

Bonus: Optimized FS access

- Particular SuperMUC Phase1 problem
  - GPFS client configuration not good for ATLAS
    - inode cache too small(1000) - delays on file access
    - G4 accesses O(1000) data files → thrashing
- cvmfs has some internal caching
  - fewer GPFS inode lookup operations
  - effect is dramatic …
    - G4 Initialization: 32mins → 5mins
    - time per event: 115s → 35s
    - both comparable to native cvmfs
    - can ramp-up phase1 usage ...

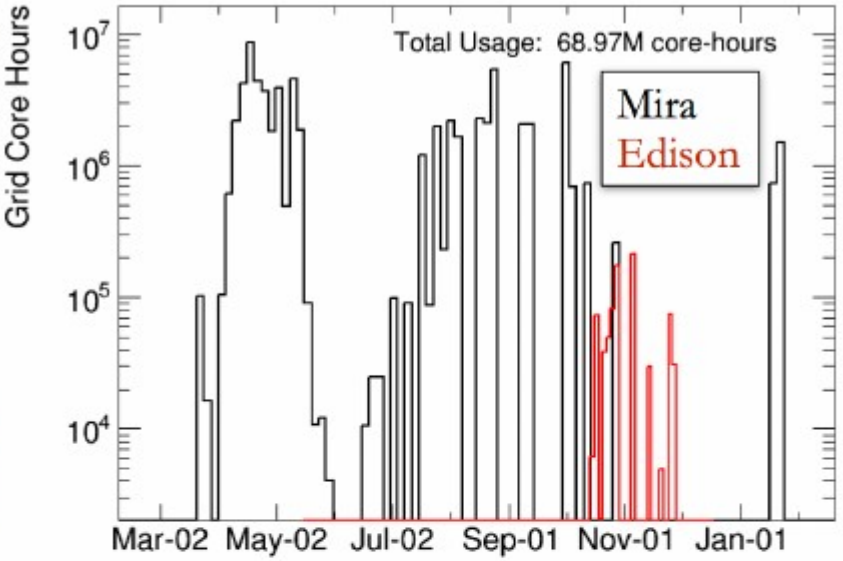
Current usage

- X86 HPC
  - SuperMUC: running 300 whole-node jobs (4800 cores)
    - usually at 300 limit.
      - often drains a little. Occasionally O(50) jobs preempted.
    - cannot delay 'proper' HPC job
    - negotiating increased limit
      - usually >1000 nodes idle
    - 10M core hours running standard production G4
  - MPI Hydra also running in production ~60 nodes
  - Titan ORNL 2M hours/month in backfill
- Event generator on PowerPC(Mira, Argonne)
  - Few multi-node jobs submitted quite manually
    - integration to prodsys ongoing, but not totally necessary

Event Generators on PPC

Argonne Opportunistic Usage

- 70M core-hours of Alpgen delivered (16B events) to ATLAS PMG in the last year. Equivalent to 5% annual grid usage.
- Normal job size is 262,144 cores, with 4 threads per core. 1.7x the Grid.
- A new request was received in mid-January for another 10M core-hours of Alpgen.
- New Alpgen version being released. Up to 10 jets possible. New requests possible. Would dwarf current usage stats. Not possible on the Grid.
- Data output averaging 1.6TB/month.
- Sherpa optimization continues, but production use has begun. 192 integrations delivered.
- Working with Eddie to add Mira usage to monitoring plots.
- Panda Integration completed. Thanks Danila.
- ProdSys Integration coming next for EVGEN jobs. Thanks Doug.



General HPC use for HEP

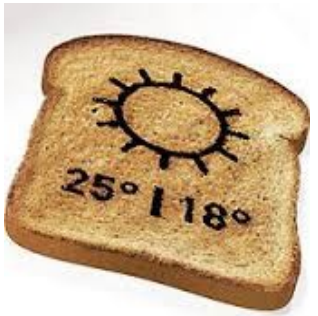
- can share of HPC replace dedicated hardware?
  - CSCS will run Tier2 on general HPC hardware
  - efficient way to provide cpu power to science
    - single facility must have cost savings
- data-intensive, user analysis workloads too?
  - potential to contribute to HPC bid and design
    - as a stake holder, HEP can ensure pledge and get backfill
- IMHO needs an attitude shift from HPC
  - in addition to HEP efforts to fit in

Hardware choices

- Like: Linux & x86 – maybe MIC too(GeantMT & Gaudi-hive)
- Agnostic about: fast network, batch system.
- Can live with: OS(SLES, BullX)
  - prefer container-based virtualization(Docker, see 'Shifter' work in US)
    - have scheduled node – should run what we want
- Unhappy: Lack of compute node disk
  - OS lives in RAM, no swap
  - no local scratch for high io or caching(cvmfs)
  - disk adds little $, and does not hurt HPC
  - 'disk' could be SSD, flash memory, RAMDISK
- Unhappy: GPU
  - we have almost no workloads to obviously benefit(maybe tracking/trigger)
  - huge effort to port, and maybe wasted when next generation comes

Policy

- Outbound connectivity
  - no self-respecting HPC code would need the Internet
    - HEP code does: Frontier, cvmfs, wget, ...
    - even toasters have Internet!



  - assumption that users and intruders are queuing up to DoS attack a litigious bank
    - destinations controlled and throttled by firewall/NAT rules

Policy(2)

- Only multi-node jobs
  - HEP has almost no need – wonderfully parallel
    - exception some evgen integration(Mira)
  - fragmentation of resources
    - scheduling question. Only short or preemptable jobs.
  - batch system load
    - only whole-node jobs implies 10k max – OK.
  - SuperMUC and Hydra accept single-node jobs
    - makes perfect sense with preemption enabled

Policy(3)

- No gateway
  - or not useful GT5, UNICORE
- Must login to headnode to submit jobs
  - key-based ssh if lucky, or securID code if in US
- HEP needs a gateway
  - integration to *automatic* production system
  - data in/out , job submit, monitor
  - real HPC users would benefit too

HPC allocation

- Very successful in the US at getting HEP allocations
  - can we learn anything
- Less so in the EU
  - official SuperMUC project initially refused
    - “does not use HPC capability” - nameless reviewer!
  - local in-house arrangements working
- Is the criteria “uses HPC” or “best science”?
  - e.g. earthquake simulation gets best scaling and flops from SuperMUC, and gets time because of this.
  - I cannot judge relative merit, but some objective panel should
    - in the same way(same place) research group is funded

Conclusion

- Initial HPC hostility overcome
  - management and admins are often positive and helpful
    - but feel inhibited by funding and computer science tradition
  - takes time, and pressure from above
    - SuperMUC, Hydra in production having made compromises
    - MIRA, Titan, Edison, … also
- HEP stake in new HPC *service* will build on this
  - challenge each policy decision, for justification
  - make clusters useful for more workloads